Virtual Face Image Generation For Illumination And Pose Insensitive Face Recognition

Wen Gao[1,2], *Shiguang Shan*[1], *Xiujuan Chai*[2], *Xiaowei Fu*[2]
[1]Institute of Computing Technology, CAS, P.O.Box 2704, Beijing, China, 100080
[2]Department of Computer Science, Harbin Institute of Technology, Harbin, China, 150001

ABSTRACT

Face recognition has attracted much attention in the past decades for its wide potential applications. Much progress has been made in the past few years. However, specialized evaluation of the state-of-the-art of both academic algorithms and commercial systems illustrates that the performance of most current recognition technologies degrades significantly due to the variations of illumination and/or pose. To solve these problems, providing multiple training samples to the recognition system is a rational choice. However, enough samples are not always available for many practical applications. It is an alternative to augment the training set by generating virtual views from one single face image, that is, relighting the given face images or synthesize novel views of the given face. Based on this strategy, this paper presents some attempts by presenting a ratio-image based face relighting method and a face re-rotating approached based on linear shape prediction and image warp. To evaluate the effect of the additional virtual face images, primary experiments are conducted using our face specific subspace method as face recognition approach, which shows impressive improvement compared with standard benchmark face recognition methods.

1. INTRODUCTION

Face recognition has attracted much attention in the past decades for its wide potential applications in commerce and law enforcement, such as mug-shot database matching, identity authentication, access control, information security, and surveillance. Much progress has been made in the past few years [1,2].

Since the 1990s, appearance based methods have been dominant researches, from which two FRT categories were derived: holistic appearance feature based and analytic local feature based. Popular methods belonging to the former paradigm include Eigenface[3], Fisherface[4]. Local Feature Analysis (LFA)[5] and Elastic Bunch Graph Matching (EBGM)[6] are typical instances of the latter category. In recent years, Eigenface, Fisherface, EBGM, Active Shape Models and Active Appearance Model (ASM/AAM)[7, 22], subspace discrimination analysis[8] and SVM[10] based approaches have attracted much attention. FERET evaluation has provided extensive comparisons of these algorithms [9].

However, face recognition remains a difficult, unsolved problem in general. The performance of almost all current face recognition systems, both best academic systems and most successful commercial systems, is heavily subject to the variations in the imaging conditions. It has been discovered by the FERET and FRVT test that pose and illumination variations are among the several bottlenecks for a practical face recognition system [9]. By far, no revolutionary practical solutions are available for these problems. However, some solutions to pose and illumination problems do have emerged including invariant feature based methods [16], 3D linear illumination subspace [4], linear object class [11], illumination and pose manifold [12], Symmetric Shape-From-Shading [8], photometric alignment [13], Quotient Image [14], illumination cones [15], Lambertian Reflectance and Linear Subspace [17], Eigen light-fields [18] and parametric linear subspace [19].

Generally, we may categorize approaches used to cope with variation in appearance into three kinds: **invariant features, canonical forms, and variation modeling** [20].

The first approach seeks to utilize features that are invariant to the changes in appearance. Examples of such representation considered by early researchers are edge maps, image intensity derivatives, and images convolved with 2D Gabor-like filers. However, Adini's empirical study had shown that "None of the representations considered is sufficient by itself to overcome image variations because of a change in the direction of illumination"[16]. Most recently, the Quotient Image [14] is reported to be invariant to illumination and may be used to recognize faces when lighting conditions change.

The second approach attempts to "normalize" away the variation in appearance, either by image transformations or by synthesizing a new image from the given image in some canonical form. Recognition is then performed using this canonical form. Examples of this approach include [8, 21].

The idea of third approach, variation modeling, is to learn, in some suitable subspace/manifold, the extent of the variation in that space/manifold. Recognition is then conducted by choosing the subspace/manifold closest to the novel image. Currently, this paradigm has been recognized as the dominant one among the three approaches[11, 12, 13, 15, 17, 19, 20].

In this paper, we investigate the possibility to augment the training set for modeling the variations by generating

0-7803-7663-3/03/$17.00 ©2003 IEEE

This paper was originally published in the Proceedings of the 2003 IEEE International Conference on Acoustics, Speech, & Signal Processing, April 6-10, 2003, Hong Kong (cancelled). Reprinted with permission.

BNSDOCID: <XP_____10650419A_I_>

virtual face images when changing lighting conditions or viewpoints. This is especially useful for applications that only limited samples per face are available for training.

The paper is organized as: In section 2 we first described briefly our works on ASM for aligning face images. Section 3 describes the ratio-image based face relighting approach, followed by virtual view prediction based on shape prediction. Our recognition approach based on Face Specific Subspaces (FSS) is presented in section 5. Experiments are set up in the last section.

2. Our Works On Feature Correspondence

Both our face relighting method and face rotating method need accurate feature correspondence. Therefore, we first describe our works on feature extraction briefly. Refer to [23], [24], [25] respectively for details of our work on face segmentation, eye localization and face shape extraction. Our face detection method, named Face Center-of-Gravity Template, is based on some observations on the configure relationship between major face organs. The eyes are then localized by growing a region window from the approximate center of the detected face and checking its characteristics. After eyes are located, we attempt to combine the ASM's local texture models and AAM's global appearance models for spare facial feature correspondence. To integrate the local profile and global appearance constraints, the subspace reconstruction residual of the global texture is exploited to evaluate the fitting degree of the current model to the novel image. And, similar to the AAMs, global texture is used to predict and tune the model parameters. Some results of our feature extraction method are shown in Fig.1.

Figure 1. Results of our feature correspondence

3. Ratio-Image Based Face Relighting for Modeling Illumination Variations

In this section, a ratio image based face relighting method is presented. The method is based on the assumptions that any face were a convex surface with a Lambertian function, that is, a face image can be described by the product of the albedo and the cosine angle between a point light source and the surface normal:

$$I(x,y) = \rho(x,y)\vec{n}(x,y)\cdot\vec{s}$$

where $\rho(x,y)$ is the albedo associated with point x,y in the image, $\vec{n}(x,y)$ is the surface normal direction associated with point x,y in the image, and $\vec{s}$ is the point light source direction and whose magnitude is the light source intensity.

Thus, our problem can be formulated as: Given a face image I_0 under normal light source, $\vec{s}_0$, we need to relight the face under other light sources, e.g. $\vec{s}_i$. To solve this problem, we present a ratio-image based method.

First, we define the ratio-image for the i^{th} face (person) under the k^{th} light source as:

$$\begin{aligned} r_{ik} &= I_{ik} / I_{i0} \\ &= (\rho_i\vec{n}_i\cdot\vec{s}_k)/(\rho_i\vec{n}_i\cdot\vec{s}_0) \\ &= (\vec{n}_i\cdot\vec{s}_k)/(\vec{n}_i\cdot\vec{s}_0) \end{aligned}$$

where ρ_i is the albedo (surface reflectance) associated with the i^{th} face, $\vec{n}_i$ is the corresponding surface normal directions, and $\vec{s}_0$ and $\vec{s}_i$ are the standard and target point light source directions respectively. Thus, we have:

$$I_{ik} = \rho_i\vec{n}_i\cdot\vec{s}_k = (\rho_i\vec{n}_i\cdot\vec{s}_0)\otimes r_{ik} = I_{i0}\otimes r_{ik}.$$

where $\otimes$ denotes Cartesian product. This means that, given the ratio-image and the standard face.image, we can relight the face to the k^{th} light source.

The ratio-image above defined is almost useless since it is only applicable to the i^{th} face. However, notice that all faces have similar 2D and 3D shapes, so we can try to first warp all faces to the same shape and then compute the ratio-image for relighting the standard face images. It is then easy to reverse warp the relit face image back to its original shape. Currently, we just warp the 2D face image to a predefined mean shape, as defined in ASM. After the warp procedure, all face images are expected to have quite similar 3D shape. Therefore, given a training set, by warp all the face images under different lighting conditions to the same shape, we can define the universal ratio-image for the k^{th} light source as the mean of all the specific ratio-image of each face in the training set:

$$R_k = \frac{1}{N}\sum_{i=1}^{N}\frac{T_{ik}}{T_{i0}} = \frac{1}{N}\sum_{i=1}^{N}R_{ik}$$

where N is the total faces in the training set, T_{ik} is the shape-free texture warped from the i^{th} face images lighted under the k^{th} light source, and T_{i0} is its corresponding texture under the standard light source.

By varying the different light source, we can get the ratio-image for each light source and/or combine them to ratio-image for any lighting conditions.

After the ratio-images are computed, any novel face image I_0 with standard lighting conditions can be relit by the following procedure:

1. Find the face in the picture and extract its 2D face shape using the method descried in Section 2;
2. Warp the face to texture T_0 according to the predefined mean shape;

BNSDOCID: <XP_____10650419A_I_>

3. Relight the face image under k-th lighting condition according to the k-th ratio-image by:

$$T_k = T_0 \otimes R_k\,;$$

4. Reverse-warp the texture T_k to its original shape to get the relit image I_k;

Fig. 2 illustrates some relighting effect of our method on the Yale face Database B (The other 9 person's faces are used to compute the 63 ratio-images for 63 light sources).

(a)Input image (b)Labeled landmarks (c)Masked image

(d) Relit images

Figure2. Ratio-image based face relighting

4. Virtual View Generation For Different Poses

Similar to face relighting, the virtual view generation problem is defined as: given a frontal view of an unknown face, generating its view under other poses. Let P_0 denote the frontal pose, P_l be another pose (e.g. right rotating 30° out of the image plane), and I_0 be the image under pose P_0. Our goal is generating its view I_l under pose P_l.

A linear regression method is exploited to solve this problem: a learning set containing pairs of the shapes of the two views under P_0 and P_l is collected, a linear mapping between them are learned and applied to any given novel frontal image to predict its shape under pose P_l. Let $\Re = \{(I_1^0, I_1^1), (I_2^0, I_2^1), \cdots, (I_m^0, I_m^1))\}$ be an image set containing pairs of the two views under P_0 and P_l, and $\Im = \{(S_1^0, S_1^1), (S_2^0, S_2^1), \cdots, (S_m^0, S_m^1))\}$ be the corresponding shape set containing pairs of the shapes for the two views in the learning set $\Re$. A linear mapping P can be learnt easily from $\Im$. So, for a given novel image I_0 under pose, P_0, its shape vector S^0 is first extracted using method in Section 2. Then, its face shape S^l viewed under pose P_l is predicted by:

$$S^1 = PS^0.$$

Then we can generate the virtual view by an image warping procedure based on S^0 and S^l. Fig.3 shows two examples of the generation results, in which the first row are the original frontal views with landmarks overlapped; the second row is the generation results.

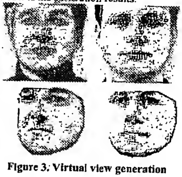


Figure 3: Virtual view generation

5. FSS-based Face Recognition

In our previous work, we have proposed a Face-Specific Subspace (FSS) based face recognition method [26]. This method is motivated, but essentially different from the traditional Eigenface. In Eigenface, each face image is represented as a point in a low dimensional face subspace shared by ***all*** faces; however, we experimentally show that one of the demerits of such a strategy is that the most discriminant features of a specific face are not accurately represented. Therefore, we propose to model each face by one individual face subspace, named Face-Specific Subspace. Distance from the face-specific subspace, that is, the reconstruction error, is then exploited as the similarity measurement for identification.

Each FSS is learnt from the training images of the specific face and represented as a 4-tuple by:

$$\Re_k = (U_k, \Psi_k, \Lambda_k, d_k),$$

where U_k is the eigenvector matrix, Λ_k is the eigenvalues, Ψ_k is the mean of the k^{th} face, and d_k is the dimension of the FSS.

Similar to DFFS in Eigenface method, the similarity of any image to a face can be measured by using the Distance From FSS (DFFSS): less DFFSS means more probability that the image belongs to the corresponding face. It can be formulated as follows: Let Γ be any input image. It can be projected to the k^{th} FSS by: $W^{(k)} = U_k^T \Phi^{(k)}$, where $\Phi^{(k)} = \Gamma - \Psi_k$. Then $\Phi^{(k)}$ can be reconstructed by: $\Phi_r^{(k)} = U_k W^{(k)}$. So, Γ's distance from k^{th} FSS (DFFSS) is computed as the following reconstruction error:

$$\varepsilon^{(k)} = \left\| \Phi^{(k)} - \Phi_r^{(k)} \right\|.$$

The DFFSS can be regarded as the similarity of the input pattern Γ to the face corresponding to the k^{th} FSS. Therefore, the following minimal distance classifier can be naturally formulated:

BNSDOCID: <XP_______10650419A__I_>

$$\Gamma \in \Omega_m \ \text{ if } \ \varepsilon^{(m)} = \min_{1 \le k \le p}\{\varepsilon^{(k)}\}.$$

6. Experiments on Yale Face Database B

To evaluate the effect of augmenting training set for face recognition, we conduct experiments on the Yale Face Database B (Refer [15] for detailed information on this face database). We choose just the frontal set in this DB, containing 640 images from 10 persons, each person has 64 frontal images under 64 different lighting conditions. To test different face recognition methods, we choose the frontal face image under the standard lighting of each person as training images, other 63 images lighted under different for testing.

Leave-one-out strategy is exploited to generate the ratio-image for each lighting configure. Then, 63 additional face images from each standard lighting face images are generated to augment the training set for the FSS method. The testing results on the 5 subsets are shown in Fig. 4.(Note: other methods tested are correlation, PCA and FaceIt3.0 system. They do not using the additional virtual images for train).

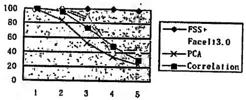


Figure 4. Performance comparison on the 5 subsets in the Yale Face Database B

From Fig.4, a significant performance improved can be observed, which obviously profits from the augment of the training set.

Acknowledgements

This research is sponsored partly by NSF of China (No.69789301), National Hi-Tech Program of China (No.2001AA114190), and YinChen Net. Co.

REFERENCES

[1] R. Brunelli and T. Poggio, "Face Recognition: Features versus Template", *TPAMI*, 15(10), pp1042-1052, 1993

[2] R.Chellappa, C.L.Wilson ect. "Human and Machine Recognition of faces: A survey", *Proc. of the IEEE*, 83(5), pp705-740, 1995.5

[3] M.Turk and A.Pentland. "Eigenfaces for Recognition" *Journal of cognitive neuroscience*, 3(1), pp71-86, 1991.1

[4] P.N.Belhumeur, J.P.Hespanha and D.J.Kriegman. "Eigenfaces vs Fisherfaces: recognition using class specific linear projection". *TPAMI*, vol.20, No.7, 1997.7

[5] P.Penev and J.Atick, "Local Feature Analysis: A General Statistical Theory for Object Representation," *Network: Computation in Neural Systems*, vol.7, pp.477-500, 1996

[6] L Wiskott, J.M.Fellous, N.Kruger and C.V.D.Malsburg, "Face Recogniton by Elastic Bunch Graph Matching", *IEEE Trans. On PAMI*, 19(7), pp775-779, 1997.7

[7] T.F.Cootes, G.J.Edwards, C.J.Taylor, "Active Appearance Models", *ECCV*, vol.2, pp484-498, 1998.

[8] W.Zhao and R.Chellappa, "Robust Image-Based 3D Face Recognition", *CAR-TR-932, N00014-95-1-0521, CS-TR-4091*, Center for Auto Research, UMD, 2000.1

[9] P.J.Phillips, H.Moon, etc. "The FERET Evaluation Methodology for Face-Recognition Algorithms", *IEEE TPAMI*, Vol.22, No.10, pp1090-1104, 2000

[10]G.Guo, S.Z.Li and K.Chan, "Face Recognition by Support Vector Machines", FG'02, pp196-201, Grenoble, 2000.3

[11]T.Vetter and T.Poggio, "Linear Object Classes And Image Synthesis From A Single Example Image", *IEEE Trans. On PAMI*, Vol.19, pp733-742, 1997

[12]H.Murase, S.Nayar, Visual Learning and recognition of 3D object from appearance, *IJCV*, 14:5-24, 1995

[13]A.Shashua, On Photometric Issues in 3D visual recognition from a single 2D Image, *International Journal of Computer Vision*, 21(1/2), 99-122, 1997

[14]A.Shashua and T.Riklin-Raviv, "The Quotient Image: Class-Based Re-Rendering And Recognition With Varying Illuminations", *IEEE Trans. on PAMI*, pp.129-139, 2001.2

[15]A.S.Georghiades, P.N.Belhumeur and D.J.Kriegman, "From Few to Many: Illumination Cone Models for Face Recognition under Differing Pose And Lighting", *IEEE TPAMI, Vol.23, No.6, pp643-660, June 2001*

[16]Y.Adini, Y. Moses, S.Ullman, Face Recognition: The Problem of Compensting for changes in illumination Direction, *IEEE TPAMI*, Vol.19, No.7, pp721-732, 1997

[17]R.Basri, D. Jacobs, Lambertian Reflectance and Linear Subspaces, ICCV2001, Beckman Institute. Vol.2, p383-390,

[18]R.Gross, I.Matthews, S.Baker, Eigen Light-Fields and Face Recognition Across Pose, Proc. of FG02

[19]K.Okada, C.Malsburg, Pose-Invariant Face Recognition with Parametric Linear Subspaces, Proc. of FG02

[20]T.Sim, T.Kanade, Combining Models and Exemplars for Face Recognition: An Illuminating Example,In Proceedings of Workshop on Models versus Exemplars in Computer Vision, CVPR 2001.

[21]P.J.Phillips, Y.Vardi, Efficient Illumination Normalization of Facial Images", PRL, 17(1996), 921-927

[22]T.F.Cootes, C.J.Taylor, D.Cooper, and J. Graham. Active shape models--their training and application. Computer vision and image understanding, 61(1): pp38-59, 1995.

[23]J.Miao, H.Zhang, W.Gao, et al., FaceTracker: A Human Face Tracking and Facial Organ Localizing System, the Proc. of the 8th ICCV, Cananda July.2001 pp.743-743,

[24]B.Cao, S.Shan, W.Gao, D.Zhao. Localizing the iris center by region growing search. Proc. of the IEEE ICME 2002.

[25]W.Wang, S.Shan, W.Gao, B.Yin, Combining Active Shape Models and Active Appearance Models For Accurate Image Interpretation, Submit to ICASSP 2003.

[26]S.Shan, W.Gao, D.Zhao, Face Identification From A Single Example Image Based On Face-Specific Subspace (FSS), Proceedings of the IEEE ICASSP2002

BNSDOCID: <XP_______10650419A__I_>

Short Papers

Online Fingerprint Template Improvement

Xudong Jiang and Wee Ser, *Senior Member, IEEE*

Abstract—This work proposes a technique that improves fingerprint templates by merging and averaging minutiae of multiple fingerprints. The weighted averaging scheme enables the template to change gradually with time in line with changes of the skin and imaging conditions. The recursive nature of the algorithm greatly reduces the storage and computation requirements of this technique. As a result, the proposed template improvement procedure can be performed online during the fingerprint verification process. Extensive experimental studies demonstrate the feasibility of the proposed algorithm.

Index Terms—Fingerprint verification, minutia set, template improvement, multiple fingerprints.

1 Introduction

An automatic fingerprint verification system matches fingerprint inputs with prestored fingerprint templates, each of which consists of a set of features extracted from a fingerprint image. Since the most reliable feature for fingerprint matching is the minutia, most current automatic fingerprint verification systems are based on minutia matching. A fingerprint template of such systems is thus a minutia set. The two most prominent kinds of minutiae are ridge ending and ridge bifurcation, which can be extracted using techniques such as those proposed in [1], [2], [3], [11]. Unfortunately, noise, inadequate contrast, and other image acquisition artifacts often make reliable minutia extraction very difficult. The resulting undesirable results include spurious minutiae being produced, valid minutiae being lost, and the minutia type (ending or bifurcation) being wrongly labeled. The employment of various image enhancement techniques [4], [5] merely alleviate these problems to a limited extent since they operate only on a single fingerprint image. Maio and Maltoni [6] implemented five different minutia extraction techniques [7], [8], [9], [10], and compared their performances. The best technique in their experiment produced 8.52 percent spurious minutiae, lost 4.51 percent genuine minutiae, and caused the type labeling error for 13.03 percent minutiae, resulting in a total error of 26.07 percent. For the other approaches, the total errors were 33.83 percent, 119.80 percent, 207.52 percent, and 216.79 percent, respectively. From this experiment, we can see that perfect minutia extraction from a single fingerprint image is a very difficult task.

Whereas the improvement of minutia extraction from a single fingerprint image is limited, multiple fingerprint images captured at different times can be used to achieve more significant improvements since the imaging conditions that cause the minutia extraction error change with time due to the changes in skin condition, climate, and on-site environment. However, improving the image quality based on multiple fingerprints is unfeasible due to the high memory and computation consumption required by processing multiple images that are not rotation and translation invariant. Instead, it is more feasible to improve the template minutia set by using multiple minutia sets of fingerprints captured at different times. If this template improving process requires only a small memory space and short computation time, it can be performed online in a fingerprint verification system, which receives fingerprint inputs of users during the day-to-day normal operation. Such online template improving functions work by merging the input data into the template database during the actual application of the fingerprint verification system.

This paper proposes an online fingerprint template improvement algorithm with which spurious minutiae can be removed, dropped minutiae be recovered, and wrongly labeled minutia type be corrected. The proposed algorithm works online during the day-to-day operation of the fingerprint verification system. As a result, users will find the system more and more reliable.

• *The authors are with the Centre for Signal Processing, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798. E-mail: {exdjiang, ewser}@ntu.edu.sg.*

Manuscript received 7 Mar. 2001; revised 11 July 2001; accepted 17 Oct. 2001. Recommended for acceptance by M. Pietikainen. For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 113757.

0162-8828/02/$17.00 © 2002 IEEE

2 Minutia Set Estimation from Multiple Minutia Sets

To extract the minutiae, the image outputted from a fingerprint sensor has to be segmented into the background (invalid fingerprint region) and the valid fingerprint region that usually covers only a part of a finger. Thus, different fingerprint images captured from the same finger usually have different (valid) fingerprint regions. A fingerprint region can be represented by a point set, which contains x- and y-coordinates of all pixels within this region. Suppose that we have M fingerprint images captured from the same finger and obtained M minutia sets $\mathbf{F}^m = \{F_k^m\}$ and fingerprint regions $\mathbf{S}^m$ by applying a minutia extraction algorithm [11], where

$$F_k^m = (x_k^m, y_k^m, \varphi_k^m, t_k^m) \tag{1}$$

is a parameter vector describing the location (x_k^m, y_k^m), the direction φ_k^m and the type t_k^m of minutia k in fingerprint m. Although the position and direction of a finger on the sensor is usually different for the various acquisitions, pose transformation can be performed using a minutia matching program [12] in order for the minutia sets and fingerprint regions of different images to be aligned. By matching the minutia sets, we can determine whether two minutiae from two different minutia sets are matched. Without losing generality, we assume that all minutia sets $\mathbf{F}^m$ and fingerprint regions $\mathbf{S}^m$ have been aligned, i.e., they are invariant to the rotation and translation of the finger, and minutiae in different sets have the same index k if and only if they are matched.

For a particular physical minutia, we obtain M' sample measurements of its parameter vector from M' different fingerprints ($M' \leq M$). Our task is to estimate an optimal parameter vector based on these M' measurements, i.e., learning from samples of experimental data. This problem could be approached in the context of minimizing a suitable cost function. If the cost function is chosen to be the negative logarithm of the likelihood function derived from the sample data, this becomes equivalent to maximum likelihood (ML) learning. By considering a generalization of the Gaussian distribution of the data with a constant variance, the ML approach leads to a cost function of the form

$$E = \sum_m |F_k^m - F_k^p|^R,$$

known as the Minkowski-R error [13], where F_k^p is the optimal representative minutia parameter vector to be estimated. When the distribution of the data is assumed to be standard Gaussian, i.e., $R = 2$, the cost function reduces to the sum-of-squares error. If a Laplacian distribution is assumed, i.e., $R = 1$, the cost function becomes the city block metric [13]. One problem of the standard sum-of-squares error is that the solution can be dominated by outliers if the distribution of data has heavy tails [14]. The use of the city block metric ($R = 1$) reduces the sensitivity to outliers but minimizing it leads to a median operation [13] on the acquired data. This is more intensive to compute compared with the simple mean operation that minimizes the sum-of-squares error. In our context of fingerprint template improvement, there are really no significant data outliers

BNSDOCID: <XP_______11094899A__I_>

since minutiae with large measurement errors cannot be matched with other corresponding minutiae with small measurement errors. This provides the motivation in using the sum-of-squares error as the cost function.

Although the biological characteristics of fingerprints ensure minutia features to be permanent and unchanging for a given finger [1], acquisition of minutiae information is affected by the skin and imaging conditions at the time of measurement and the exact manner the finger was making contact with the sensor. As a result, the measured minutia parameter inevitably changes with time and the measurements F_k^m can thus be seen as a temporal sequence of data. As such, the machine-learning task could be viewed as a problem of regression estimation, i.e., function or model learning. There are a number of well-developed approaches in the literature for these problems, for example, LPC [15], Kalman filtering [16], Hidden Markov model [17], MCMC methods [18] and EM-C algorithm [19]. However, changes to minutia parameters may occur abruptly with these changes being maintained for quite a long time due to the skin nature and human's habits. F_k^m is thus typically an abrupt rather than a smooth function of m, which makes it difficult to apply the above-mentioned function learning approaches. Furthermore, fingerprint samples for a particular user are not collected in even time intervals; duration between two subsequent presentations of a finger to the system may vary between several minutes to several months. This again makes the above-mentioned approaches unsuitable.

Having considered the above factors and the computational efficiency required for an online application, we employ the weighted least-squares with predetermined weights as the learning rule. The weights are chosen based on the nature of minutia set series, the objective of the integration of the multiple minutia sets, and the computation efficiency. For instance, a higher weight should be assigned to the registered template than the query fingerprint received in the verification process since the original template obtained during the registration phase is generally more reliable than the input minutia sets obtained during the day-to-day verification process. More recent fingerprint inputs should also be assigned with higher weights than earlier ones since the integration of the multiple minutia sets is aimed at increasing the reliability of future matching process. The weights will be chosen in the next section based on these desired factors and the computational resources required.

The estimation errors for all minutiae k of all minutia sets m are expressed as

$$e_k^m = F_k^m - F_k^P, \quad \text{for } \forall (k,m) F_k^m \in \mathbf{F}^m. \tag{2}$$

The estimated minutia F_k^P is obtained by minimizing the weighted sum of the squared errors

$$\begin{aligned}\sum_{m, F_k^m \in \mathbf{F}^m} w_k^m (e_k^m)^2 &= \sum_{m, F_k^m \in \mathbf{F}^m} w_k^m (F_k^m - F_k^P)^2 \\ &\Rightarrow \text{Minimum}, \quad \text{for } \forall k, F_k^m \in \mathbf{F}^m,\end{aligned} \tag{3}$$

where w_k^m are predetermined weights. Based on this criterion, it is straightforward to obtain

$$F_k^P = \frac{1}{\sum_{m, F_k^m \in \mathbf{F}^m} w_k^m} \cdot \sum_{m, F_k^m \in \mathbf{F}^m} w_k^m F_k^m; \quad \text{for } \forall k, F_k^m \in \mathbf{F}^m. \tag{4}$$

The above estimated minutia parameter F_k^P generally has better accuracy than F_k^m since it is a weighted arithmetic average over all matched minutiae. As a result, wrongly labeled minutiae type can be statistically corrected during the averaging process.

If all estimated minutiae by (4) are collected in the estimated template, a template synthesis will be performed. If we match an input fingerprint with this template, we will face the problem of matching a partial input fingerprint with a much larger full fingerprint. False nonmatch rate may decrease but false match rate may increase simultaneously, or some matching criteria have to be changed by compromise. Further, template synthesis can be simply offline performed by enrolling several fingerprints for each finger and there is no point in online synthesizing the templates. The problem of the template synthesis was addressed in [20]. This work is not aimed at solving the problem that the template represents only a partial fingerprint, but aimed at improving the quality of the template online, i.e., reducing the minutia extraction error. Thus, our estimated template is restricted to an estimated fingerprint region $\mathbf{S}^P$, which can be chosen to be one of the M original fingerprint regions $\mathbf{S}^m$ or be determined by the synthesized fingerprint region in case the template synthesis is performed beforehand in the registration phase.

Our estimated minutia set $\mathbf{F}^P = \{F_k^P | (x_k^P, y_k^P) \in \mathbf{S}^P\}$ contains all minutiae in the region $\mathbf{S}^P$ extracted from the M fingerprints. As a result, genuine minutiae that are not extracted from some fingerprints can be recovered in the estimated minutia set $\mathbf{F}^P$ if they are successfully extracted from some other fingerprints. However, any spurious minutia extracted from any fingerprint is also transferred to the estimated minutia set if it is located within $\mathbf{S}^P$. Therefore, a technique has to be developed to identify the spurious minutiae of the estimated template $\mathbf{F}^P$.

If an estimated minutia k is successfully extracted from fingerprint m, a certainty level $c_k^m = 1$ is defined. If this minutia fails to be extracted from fingerprint m but its location is within this fingerprint region, a certainty level $c_k^m = 0$ is defined. However, if the region of fingerprint m does not cover this minutia, no information about the reliability of minutia k is provided by fingerprint m. Thus, the reliability of each estimated minutia is described by M certainty levels defined by

$$\begin{aligned}&c_k^m = \\ &\begin{cases} 1, & \text{if } F_k^m \in \mathbf{F}^m \\ 0, & \text{if } F_k^m \notin \mathbf{F}^m \wedge (x_k^P, y_k^P) \in \mathbf{S}^m, \\ \text{unknown}, & \text{if } (x_k^P, y_k^P) \notin \mathbf{S}^m \end{cases} \\ &\text{for } \forall k, F_k^P \in \mathbf{F}^P, m = 1, 2, \ldots M,\end{aligned} \tag{5}$$

Similar to the calculation of the estimated minutia parameter in (4), a certainty level c_k^P of the estimated minutia k can be estimated by the weighted average of c_k^m over all fingerprints whose regions cover the minutia k

$$c_k^P = \frac{1}{\sum_{m, (x_k^P, y_k^P) \in \mathbf{S}^m} w_k^m} \cdot \sum_{m, (x_k^P, y_k^P) \in \mathbf{S}^m} w_k^m c_k^m, \quad \text{for } \forall k, F_k^P \in \mathbf{F}^P. \tag{6}$$

For authenticating a future input fingerprint, only the minutiae whose certainty levels are equal to or higher than a threshold Cv, $0 < Cv < 1$, will be used in the matching. This means that minutiae with certainty levels lower than Cv are regarded as spurious minutiae, which must still remain in the template for the further template improvement in the future or can be removed from the template to reduce the template size if further template improvement is not conducted.

If we choose equal weights, the estimation of a minutia based on (4) and (6) is simplified to

$$F_k^P = \frac{1}{M_k} \cdot \sum_{m, F_k^m \in \mathbf{F}^m} F_k^m, \quad \text{for } \forall k, F_k^m \in \mathbf{F}^m, \tag{7}$$

$$c_k^P = \frac{1}{N_k} \cdot \sum_{m, (x_k^P, y_k^P) \in \mathbf{S}^m} c_k^m, \quad \text{for } \forall k, F_k^P \in \mathbf{F}^P, \tag{8}$$

where M_k is the number of fingerprints from which minutia k is extracted and N_k is the number of fingerprints that cover the position of minutia k. In this case, we see that it is the minutia occurrence frequency that is used to recover dropped minutiae and remove spurious minutiae.

BNSDOCID: <XP_______11094699A__I_>

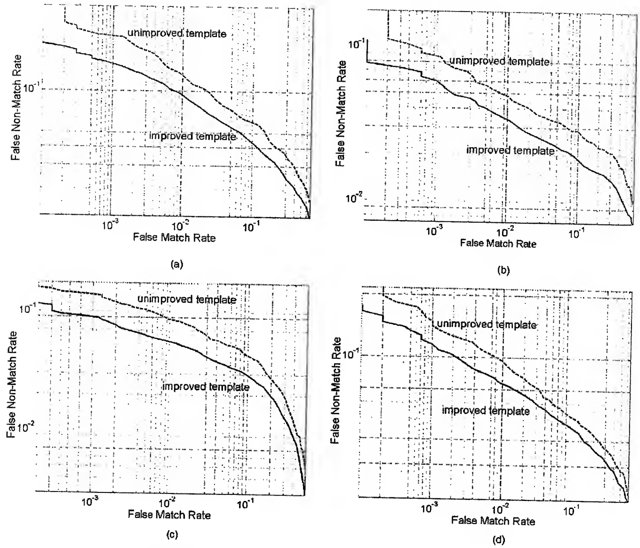


Fig. 1. ROC curves on (a) DB1_a, (b) DB2_a, (c) DB3_a, and (d) DB4_a.

3 Online Fingerprint Template Improvement

It is not user-friendly to capture a number of fingerprints of the same finger at long intervals in the registration phase. However, during the verification operation of a fingerprint verification system, input fingerprints are successively received and compared with the templates. If an input fingerprint is successfully matched with a template, these two fingerprints are verified to have originated from the same finger. Therefore, input fingerprints can be used to improve the matched template online during the day-to-day operation of the fingerprint verification system. However, to use (4) and (6) to improve a template, all matched input minutia sets and fingerprint regions need to be stored and an arithmetic average over all matched minutia sets need to be calculated for every template update. This requires a large storage space and significant computation time. However, storage space and verification time are often serious constraints, especially in stand-alone application. To reduce the storage space and processing time requirements, we need a simplified fingerprint region representation and a recursive algorithm.

It is not difficult to simplify the fingerprint region representation. A polygon represented by a few points can be used to approximate a fingerprint region [21] and to determine whether a minutia point of another fingerprint is located within this region. It is easy to prove that a point with location (x, y) is within a polygon represented by L points (x_j, y_j), $j = 1, 2, \ldots L$, if and only if:

$$A_j x + B_j y + C_j < 0, \quad \text{for all } j, j = 1, 2, \ldots, L, \tag{9}$$

where $A_j = y_j - y_{j-1}$, $B_j = x_{j+1} - x_j$, and $C_j = -A_j x_j - B_j y_j$ with $(x_{L+1}, y_{L+1}) = (x_1, y_1)$.

A recursive algorithm that implements the weighted averaging in (4) and (6) can be derived by choosing the weights properly. As mentioned earlier, the finger skin and imaging condition changes with time and the template improvement is aimed at increasing the reliability of future matching processes. Thus, the more recent fingerprint inputs should be assigned higher weights than earlier ones. In addition, a fingerprint image is usually captured with much more caution during the registration process compared with those acquired during the day-to-day verification. Therefore, a higher weight should be assigned to the original registered template. We thus choose a power series α^n to weight the fingerprint sequence and another constant λ to distinguish the weights of input fingerprints from that of the original template fingerprint.

Let $F_i^T(N)$ denote the improved template minutia by using an original template minutia $F_i^P(0)$ and N input minutiae $F_i^I(n)$ of N input fingerprints, $n = 1, 2 \ldots$ N, where fingerprint n is received

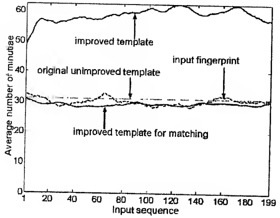


Fig. 2. Average numbers of minutiae.

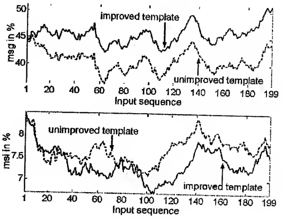


Fig. 3. Average matching scores.

earlier than fingerprint $n+1$. Without losing generality, (4) can be rewritten as

$$F_k^P(N) = \alpha^N F_k^P(0) + \lambda \sum_{n=0}^{N-1} \alpha^n F_k^I(N-n), \tag{10}$$

with the condition of

$$\alpha^N + \lambda \sum_{n=0}^{N-1} \alpha^n = 1. \tag{11}$$

The power series coefficients $\alpha^n (\alpha < 1)$ weight the more recent fingerprint entries more heavily than earlier ones, while the constant $\lambda (\lambda < 1)$ scales the weights of input fingerprints with respect to the original template. By choosing $\lambda = 1 - \alpha$, it is not difficult to prove that condition (11) holds independent of the values of α and N. Thus, we can use the same value of α in (10) for different number of entries N, i.e., we can have

$$F_k^P(N+1) = \alpha^{N+1} F_k^P(0) + \lambda \sum_{n=0}^{N} \alpha^n F_k^I(N+1-n). \tag{12}$$

From (10) and (12) with $\lambda = 1 - \alpha$, it is straightforward to obtain

$$F_k^P(N+1) = \alpha F_k^P(N) + (1-\alpha) F_k^I(N+1). \tag{13}$$

Equation (13) is the recursive template minutia parameter update formula where $F_k^P(N)$ is the old template minutia parameter and $F_k^P(N+1)$ the new template minutia parameter after the fingerprint verification system receives a new entry $F_k^I(N+1)$. If an input minutia $F_k^I(N+1)$ is located within the template fingerprint region, but there is no old template minutia $F_k^P(N)$ matched with it, this input minutia will be merged into the template, i.e., $F_k^P(N+1) = F_k^I(N+1)$.

In a similar way, a recursive certainty level update formula can be easily derived from (6) as

$$c_k^P(N+1) = \alpha c_k^P(N) + (1-\alpha) c_k^I(N+1), \tag{14}$$

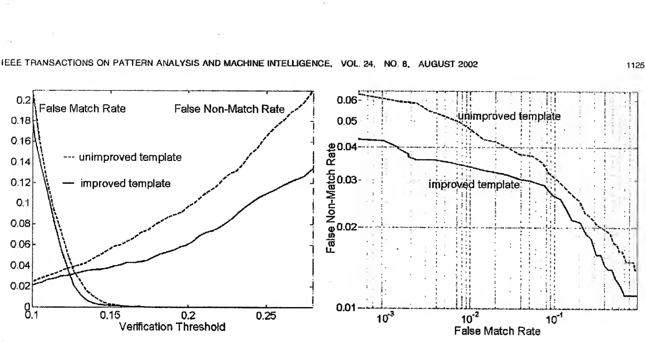
where $c_k^P(N)$ is the old certainty level of template minutia k and $c_k^P(N+1)$ the new certainty level of template minutia k after the system receives a new entry with certainty leve $c_k^I(N+1)$.

It is worth noting that falsely matched input fingerprints will have an adverse effect on the template improvement process. To reduce the probability of this happening, an input fingerprint is used to update the matched template only if their matching score ms is higher than a threshold Mu that is set to be larger than the verification threshold Mv of a fingerprint verification system. Furthermore, we could limit the shortest time interval Ti between two successive updates of a template to prevent any intentional repeated abuse of the template update process. This way, we can diminish the negative effects of the online fingerprint template improvement procedure.

The proposed online fingerprint template improvement algorithm is summarized as follows:

1. Users enroll for the fingerprint verification system. For each enrolled fingerprint, a fingerprint region S^p is segmented out and represented by L points and a minutia set $\{F_k^P\}$ is extracted. A certainty level $c_k^P = 1$ is initialized for each extracted minutia. Template update time tu is initialized to be the current time tc.
2. The fingerprint verification system waits until an input fingerprint is received.
3. Segment out the fingerprint region and represent it with L points. Extract input minutia set $\{F_k^I\}$. Match it with each template $\{F_k^P | c_k^P \geq Cv\}$. If the maximal matching score $ms \leq Mv$, reject this input and go to Step 2, otherwise output the corresponding finger ID.
4. Read the current time tc. If $ms \leq Mu\ (Mu > Mv)$ or $(tc - tu) < Ti$, go to Step 2, otherwise $tu = tc$.
5. The matched template $\{F_k^P\}$ is updated as follows:
   a. For all matched template minutiae, F_k^P and c_k^P are updated by
   $$\alpha F_k^P + (1-\alpha) F_k^I \Rightarrow F_k^P$$
   and $\alpha c_k^P + 1 - \alpha \Rightarrow c_k^P$.
   b. Find all unmatched template minutiae located within the input fingerprint region by using (9). The certainty levels of these minutiae are updated by $\alpha c_k^P \Rightarrow c_k^P$.
   c. The certainty levels of other template minutiae (i.e., those located outside the input fingerprint region) are unchanged, i.e., $c_k^P \Rightarrow c_k^P$.
   d. Find all unmatched input minutiae located within the template fingerprint region by using (9). Merge these minutiae into the template, i.e., $F_k^I \Rightarrow F_k^P$ with $c_k^P = 1 - \alpha$.
   e. Remove all template minutiae whose certainty levels are lower than a threshold $Cu (0 < Cu < 1 - \alpha < Cv < 1)$ to limit the enlargement of the template size. Store the updated template and go to Step 2.

The above proposed online fingerprint template improvement algorithm updates a fingerprint template using a recursive algorithm that implements a weighted averaging over all matched fingerprints. It increases the precision of minutia parameter, corrects



(a) (b)

Fig. 4. (a) False match rate and false nonmatch rate versus threshold and (b) ROC curves on DB_T.

wrongly labeled minutia type, recovers dropped minutiae (from $c_k^p < Cv$ to $c_k^p \geq Cv$) and removes spurious minutiae (from $c_k^p \geq Cv$ to $c_k^p < Cv$). Furthermore, the algorithm accords greater importance to more recently acquired fingerprints so that it weakens the effect of older skin and imaging conditions while strengthening recent ones. The recursive algorithm works only on the current input fingerprint and the stored template. This greatly reduces the processing time and storage space requirements and, therefore, enables our proposed approach to be online employed.

4 Experimental Studies

For a meaningful performance evaluation of our online fingerprint template improvement algorithm, the test database should contain not only a large number of finger IDs, but also a large number of sample fingerprints per finger. Unfortunately, it is very difficult to find or build up such a database. Thus, we decided to conduct experiments with two different kinds of databases. The first experiment used the FVC2000 [22] databases that contain a large number of finger IDs and eight sample fingerprints per finger. The second experiment used a database collected by us that contains only 12 finger IDs but 200 sample fingerprints per finger. The algorithm parameters chosen for both experiments were: $L = 8$, $\alpha = 0.8$, $C_u = 0.15$, $C_v = 0.5$, $Mu = Mv = 0.25$, and $Ti = 0$.

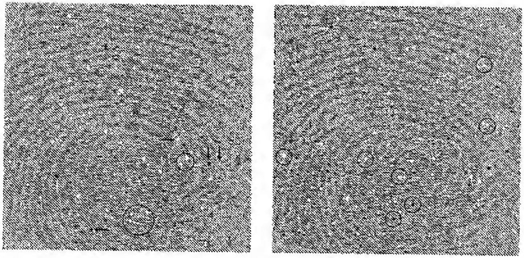
There are four databases, DB1_a, DB2_a, DB3_a, and DB4_a, used in FVC2000. Each database contains 100 finger IDs and eight fingerprints per finger (800 fingerprints in all). Let $\mathbf{F}_j^i$ denote the minutia set extracted from the ith sample fingerprint of the jth finger, $i = 1, 2, \ldots 8$, $j = 1, 2, \ldots 100$. To test the verification performance with the unimproved template, each template $\mathbf{T}_j^i = \mathbf{F}_j^i$ was matched against the inputs $\mathbf{F}_j^k$ $(i < k \leq 8)$ to obtain the genuine matching scores and each template $\mathbf{T}_j^1 = \mathbf{F}_j^1$ was matched against the inputs $\mathbf{F}_k^1$ $(j < k \leq 100)$ to obtain the impostor matching scores. Let $\mathbf{T}_j^i(k)$ denote the improved template by using the original template $\mathbf{F}_j^i$ and six inputs $\mathbf{F}_j^m$ $(1 \leq m \leq 8, m \neq i, m \neq k, i < k \leq 8)$. To test the verification performance with the improved template, each improved template $\mathbf{T}_j^i(k)$ was matched against the input $\mathbf{F}_j^k$ to obtain the genuine matching score and each template $\mathbf{T}_j^1(2)$ was matched against the inputs $\mathbf{F}_k^1$ $(j < k \leq 100)$ to obtain the impostor matching scores. In both tests, a total of $((8 \times 7)/2) \times 100 = 2,800$ genuine matches and $(100 \times 99)/2 = 4,950$ impostor matches were performed on each database. These numbers of matches are the same as that used in the FVC2000 [22]. Fig. 1 illustrates the ROC curves on the four FVC2000 databases. These ROC curves on the four different databases consistently show that our template improvement algorithm causes a significant improvement in the verification accuracy.

In the second experiment, a Veridicom CMOS sensor of size 300×300 pixels was used to capture fingerprints. Twelve untrained users were enrolled by capturing one template fingerprint per user. Each of these 12 users was asked to represent the enrolled finger to the system to produce input fingerprints several times (no more than five times) every working day until 199 input fingerprints per user were received. Each input fingerprint was matched online with the 12 templates and used to update (online improve) the template that had the maximal matching score if this matching score was higher than Mu. In this way, 2,400 fingerprints were collected in database DB_T and the numbers of genuine matches and imposter matches are $199 \times 12 = 2,388$ and $199 \times 11 \times 12 = 26,268$, respectively.

To show the template improvement progress clearly against the fingerprint input sequence, we averaged the results over the 12 users. Fig. 2 plots the average numbers of minutiae of the input fingerprints, the original templates, and the improved templates as well as the average numbers of improved template minutiae that were valid for matching. From Fig. 2, the improved template size increased with the first few fingerprint inputs and then stabilized at around twice the original unimproved template size. However, Fig. 2 also shows that the improved template had averaged fewer valid minutiae for matching than the original template. It means that, in most cases, the template improvement process removed more spurious minutiae compared with recovering dropped minutiae. This is because our minutia extraction algorithm, like most other minutia extraction approaches [6], usually produces more spurious minutiae than dropped minutiae. This experiment also tells us that the improved verification performance is not due to the increased number of minutiae in the improved template but the improved template quality.

Fig. 3 illustrates the average genuine matching scores msg and the average imposter matching scores msi against the fingerprint input sequence. Fig. 3 clearly shows that the improved template produced not only higher genuine matching scores but also lower imposter matching scores than the unimproved template. Obviously, the higher genuine matching score is due to the template improvement process. The lower imposter matching score also does not surprise us as the template improvement process reduced the number of spurious minutiae and, thus, lowered the impostor matching score. Both the higher genuine matching and the lower imposter matching scores improved the fingerprint verification accuracy, as shown in Fig. 4.

BNSDOCID: <XP_______11094699A_I_>



(a) (b)

Fig. 5. (a) Original template minutiae and (b) improved template minutiae.

The average time taken by the fingerprint verification process (one minutia set extraction and one matching) was 0.147 seconds and the average time taken by the template improvement process was only 0.0003 seconds (for a Pentium III-733 MHz PC). Our template improvement algorithm only decelerates the fingerprint verification system by a negligible 0.2 percent.

Fig. 5a shows an original template minutia set while Fig. 5b shows the improved template minutia set (valid for matching, i.e., $\{F_k^P | c_k^P \geq 0.5\}$), generated using the template minutia set in Fig. 5a and 26 other input minutia sets. In these two figures, white dots represent endings, dark dots represent bifurcations, while white short lines represent the minutia directions. There are 30 minutiae in Fig. 5a and 32 minutiae in Fig. 5b. After the template improvement, five spurious minutiae were removed (see circles in Fig. 5a), seven dropped minutiae were recovered (see circles in Fig. 5b) and four minutiae had their type relabeled (see the arrows in Fig. 5a).

5 Conclusions

In this work, an online fingerprint template improvement algorithm is proposed. The proposed algorithm improves the reliability of a fingerprint template by using weighted averaging over all matched fingerprints that a fingerprint verification system receives. It reduces minutia extraction errors, such as spurious minutiae, dropped minutiae, and wrongly labeled minutia type, which are difficult to avoid using only a single fingerprint image. Furthermore, the template is gradually changed to reflect changes in the finger skin and imaging conditions by weighting recent fingerprints more heavily. A recursive algorithm minimizes the storage space and computation requirements of the template improvement process. As a result, the proposed fingerprint template improvement process can be performed online during the day-to-day operation of a fingerprint verification system. Extensive experimental studies demonstrate that the proposed online template improvement technique significantly increase the verification accuracy at a negligible cost in time. One problem of this technique is the enlargement of the template size. However, the improved template size is well limited to within twice the size of the original template.

References

[1] A.K. Jain, L. Hong, and R. Bolle, "On-Line Fingerprint Verification," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 4, pp. 302-314, Apr. 1997.

[2] A.K. Jain, L. Hong, S. Pankanti, and R. Bolle, "An Identity-Authentication System Using Fingerprints," *Proc. IEEE*, vol. 85, no. 9, pp. 1365-1388, Sept. 1997.

[3] X.D. Jiang, W.Y. Yau, and W. Ser, "Minutiae Extraction by Adaptive Tracing the Gray Level Ridge of the Fingerprint Image," *Proc. IEEE Sixth Int'l Conf. Image Processing*, vol. 2, pp. 852-856, 1999.

[4] L. Hong, Y. Wan, and A.K. Jain, "Fingerprint Image Enhancement: Algorithm and Performance Evaluation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 777-789, Aug. 1998.

[5] X.D. Jiang, W.Y. Yau, and W. Ser, "Fingerprint Image Processing for Automatic Verification," *Proc. IEEE Second Int'l Conf. Information, Comm. & Signal Processing*, 1999.

[6] D. Maio and D. Maltoni, "Direct Gray-Scale Minutiae Detection in Fingerprints," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 1, pp. 27-39, 1997.

[7] R.M. Stock and C.W. Swonger, "Development and Evaluation of a Reader of Fingerprint Minutiae," Technical Report CAL No. XM-2478-X-1: 13-17, Cornell Aeronautical Laboratory, 1969.

[8] B. Moayer and K. Fu, "A Tree System Approach for Fingerprint Pattern Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 8, no. 3, pp. 376-388, 1986.

[9] M.R. Verma, A.K. Majumdar, and B. Chatterjee, "Edge Detection in Fingerprints," *Pattern Recognition*, vol. 20, no. 5, pp. 513-523, 1987.

[10] L. O'Gorman and J.V. Nickerson, "An Approach to Fingerprint Filter Design," *Pattern Recognition*, vol. 22, no. 1, pp. 29-38, 1989.

[11] X.D. Jiang, W.Y. Yau, and W. Ser, "Detecting the Fingerprint Minutiae by Adaptive Tracing the Gray Level Ridge," *Pattern Recognition*, vol. 34, no. 5, pp. 999-1013, May 2001.

[12] X.D. Jiang and W.Y. Yau, "Fingerprint Minutiae Matching Based on the Local And Global Structures," *Proc. 15th Int'l Conf. Pattern Recognition*, vol. 2, pp. 1042-1045, 2000.

[13] Ch.M. Bishop, *Neural Networks for Pattern Recognition*. Oxford Univ. Press, 1995.

[14] P.J. Huber, *Robust Statistics*. New York: John Wiley, 1981.

[15] L. Rabiner and J.B.-H. Juang, *Fundamentals of Speech Recognition*. Prentice-Hall, 1993.

[16] Y. Bar-Shalom and T. Fortmann, *Tracking and Data Association*. Academic Press, 1988.

[17] G.E. Kopec and P.A. Chou, "Document Image Decoding Using Markov Source Models," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, no. 6, pp. 602-617, June 1994.

[18] P. Magni, R. Bellazzi, and G.D. Nicolao, "Bayesian Function Learning Using MCMC Methods," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 12, pp. 1319-1331, Dec. 1998.

[19] B. North, A. Blake, M. Isard, and J. Rittscher, "Learning and Classification of Complex Dynamics," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 9, pp. 1016-1034, Sept. 1997.

[20] K.A. Toh, W.Y. Yau, X.D. Jiang, T.P. Chen, J. Lu, and E. Lim, "Minutiae Data Synthesis for Fingerprint Identification Application," *Proc. IEEE Int'l Conf. Image Processing*, 2001.

[21] R.C. Gonzalez and R.E. Woods, *Digital Image Processing*. Addison Wesley, 1993.

[22] D. Maio, D. Maltoni, R. Cappelli, J.L. Wayman, and A.K. Jain, "FVC2000: Fingerprint Verification Competition," *Proc. 15th Int'l Conf. Pattern Recognition*, 2000.

BNSDOCID: <XP_______11094699A_I_>

PERGAMON

Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT

Pattern Recognition 37 (2004) 1533–1542

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/patcog

Biometric template selection and update: a case study in fingerprints

Umut Uludag[a], Arun Ross[b,*], Anil Jain[a]

[a] Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824, USA
[b] Department of Computer Science and Electrical Engineering, West Virginia University, Morgantown, WV 26506, USA

Received 23 June 2003; accepted 13 November 2003

Abstract

A biometric authentication system operates by acquiring biometric data from a user and comparing it against the template data stored in a database in order to identify a person or to verify a claimed identity. Most systems store multiple templates per user in order to account for variations observed in a person's biometric data. In this paper we propose two methods to perform automatic template selection where the goal is to select prototype fingerprint templates for a finger from a given set of fingerprint impressions. The first method, called DEND, employs a clustering strategy to choose a template set that best represents the intra-class variations, while the second method, called MDIST, selects templates that exhibit maximum similarity with the rest of the impressions. Matching results on a database of 50 different fingers, with 200 impressions per finger, indicate that a systematic template selection procedure as presented here results in better performance than random template selection. The proposed methods have also been utilized to perform automatic template update. Experimental results underscore the importance of these techniques.

© 2003 Pattern Recognition Society. Published by Elsevier Ltd. All rights reserved.

Keywords: Biometrics; Template selection; Template update; Fingerprints; Clustering; Prototype; Template aging

1. Introduction

A biometric authentication system uses the physiological (fingerprints, face, hand geometry, iris) and/or behavioral (voice, signature, keystroke dynamics) traits of an individual to identify a person or to verify a claimed identity [1]. A typical biometric system operates in two distinct stages: the enrollment stage and the authentication stage. During enrollment, a user's biometric data (e.g., fingerprints) is acquired and processed to extract a feature set (e.g., minutiae points) that is stored in the database. The stored feature set, labelled with the user's identity, is referred to as a template. In order to account for variations in the biometric data of a user, multiple templates corresponding to each user may be stored. During authentication, a user's biometric data is once again acquired and processed, and the extracted feature set is matched against the template(s) stored in the database in order to identify a previously enrolled individual or to validate a claimed identity. The matching accuracy of a biometrics-based authentication system relies on the stability (permanence) of the biometric data associated with an individual over time. In reality, however, the biometric data acquired from an individual is susceptible to changes due to improper interaction with the sensor (e.g., partial fingerprints, change in pose during face-image acquisition), modifications in sensor characteristics (e.g., optical vs. solid-state fingerprint sensor), variations in environmental factors (e.g., dry weather resulting in faint fingerprints) and temporary alterations in the biometric trait itself (e.g., cuts/scars on fingerprints). In other words, the biometric measurements tend to have a large intra-class variability. Thus, it is possible for the stored template data to be significantly different

* Corresponding author. Tel.: +1-304-293-0405; fax: +1-304-293-8602.

E-mail addresses: uludagum@cse.msu.edu (U. Uludag), ross@csee.wvu.edu (A. Ross), jain@cse.msu.edu (A. Jain).

0031-3203/$30.00 © 2003 Pattern Recognition Society. Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2003.11.012

BNSDOCID: <XP_______4508402A__I_>

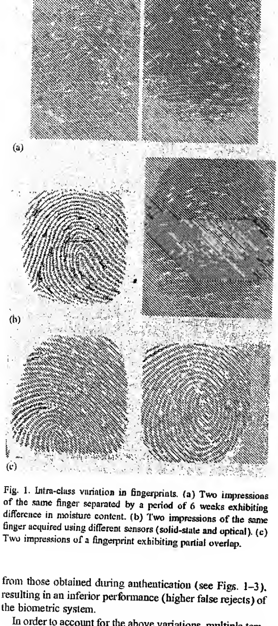


Fig. 1. Intra-class variation in fingerprints. (a) Two impressions of the same finger separated by a period of 6 weeks exhibiting difference in moisture content. (b) Two impressions of the same finger acquired using different sensors (solid-state and optical). (c) Two impressions of a fingerprint exhibiting partial overlap.

from those obtained during authentication (see Figs. 1–3), resulting in an inferior performance (higher false rejects) of the biometric system.

In order to account for the above variations, multiple templates, that best represent the variability associated with a user's biometric data, should be stored in the database. For example, one could store multiple impressions pertaining to different portions of a user's fingerprint in order to deal with the problem of partially overlapping fingerprints. Similarly, a user's face image acquired from multiple viewpoints may be stored in order to account for variations in a person's pose. There is a tradeoff between the number of templates, and the storage and computational overheads introduced by multiple templates. For an efficient functioning of a biometric system, the process of template selection has to be automated. However, there is limited literature dealing with the problem of automatic template selection in a biometric system.[1] In this paper we propose techniques to perform automatic *template selection*. The methods presented here attempt to represent the *variability* as well as the *typicality* observed in a user's biometric data. The proposed methods have also been utilized to perform automatic *template update*. Our experimental results indicate the importance of adopting a formal procedure to perform template selection and update. Although we consider a fingerprint-based biometric system as our test-bed, the techniques presented in this paper may be applied to other types of biometric traits (such as face and hand geometry) as well.

The rest of the paper is organized as follows. In Section 2 the two methods used to perform template selection have been described; in Section 3 the methodologies used to perform template update have been explained; Section 4 describes the experiments conducted to study the effectiveness of the proposed techniques; Section 5 summarizes the results of this work and provides future directions for research.

2. Template selection

The problem of template selection with regard to fingerprints may be posed as follows: Given a set of N fingerprint images corresponding to a single finger, select K templates that 'best' represent the variability as well as the typicality observed in the N images, $K < N$. Currently, we assume that the value of K is predetermined. This systematic selection of templates is expected to result in a better performance of a fingerprint matching system compared to a random selection of K templates out of the N images.

It is important to note that template selection is different from template update. The term template update is used to refer to one of the following situations: (i) *Template aging*: Certain biometric traits of an individual vary with age. The hand geometry of a child, for example, changes rapidly during the initial years of growth. To account for such changes, old templates have to be regularly replaced/augmented with newer ones. The old templates are said to undergo aging. (ii) *Template improvement*: A previously existing template may be modified to include information obtained at a more recent time instance. For example, minutiae points may be added to, or deleted/modified from the template of a fingerprint, based on information observed in recently acquired impressions [3–5]. As another example, Liu et al. [6] update the eigenspace in a face recognition system via decay parameters that control the influence of old and new training samples of face images. Thus, template selection refers to

[1] Template selection has been studied in the context of other pattern recognition problems (see Ref. [2], for example).

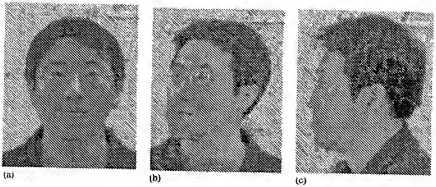


Fig. 2. Intra-class variation associated with an individual's face image. Due to change in pose, an appearance-based face recognition system will not be able to match these three images successfully, although they belong to the same individual [12].

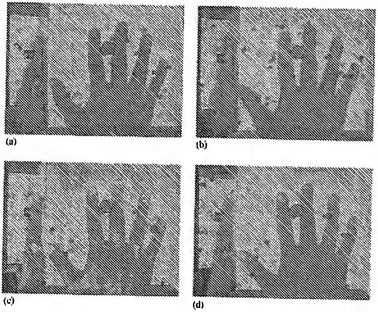


Fig. 3. Hand images of an individual acquired over a period of 4 years using the same hand reader [13].

the process by which prototype templates are chosen from a given set of samples, whereas template update refers to the process by which existing templates are either replaced or modified. In the next section, we present techniques for template update, based on the template selection methods presented in this section.

We propose the following two methods for template selection:

Method 1 (DEND): In this method, the N fingerprint impressions corresponding to a user are grouped into K clusters, such that impressions within a cluster are more similar than impressions from different clusters. Then for each cluster, a prototype (representative) impression that typifies the members of that cluster is chosen, resulting in K template impressions. This technique, therefore, selects prototypes that represent the variability observed in the impressions.

To perform clustering, it is required to compute the (dis)similarity between fingerprint impressions. This measure of (dis)similarity is obtained by matching the minutiae point sets of the fingerprint impressions. Our matching

BNSDOCID: <XP_______4508402A__I_>

algorithm is based on an elastic string matching technique [7], and it outputs a distance score indicating the dissimilarity of the minutiae sets being compared. We use a simple matching algorithm since our goal is to perform template selection, regardless of the characteristics of the matching algorithm.

Since our representation of the N fingerprint impressions is in the form of a $N \times N$ dissimilarity matrix instead of a $N \times d$ pattern matrix (d is the number of features), we use hierarchical clustering [8]. In particular, we use an agglomerative complete link clustering algorithm. The output of this algorithm is a dendrogram which is a binary tree, where each terminal node corresponds to a fingerprint impression, and the intermediate nodes indicate the formation of clusters (see Fig. 4).

The template set T, $|T| = K$, is selected as follows:

Step 1: Generate the $N \times N$ dissimilarity matrix, M, where entry (i,j), $i,j \in \{1,2,\ldots,N\}$ is the distance score between impressions i and j.

Step 2: Apply the complete link clustering algorithm on M, and generate the dendrogram, D. Use the dendrogram D to identify K clusters.

Step 3: In each of the clusters identified in step 2, select a fingerprint impression whose average distance from the rest of the impressions in the cluster is minimum. If a cluster has only 2 impressions, choose any one of the two impressions at random.

Step 4: The impressions selected in step 3 constitute the template set T.

In Step 2, the algorithm automatically determines the threshold distance to cut the dendrogram and identify exactly K clusters. For example, for the dendrogam given in Fig. 4, this distance is determined to be 644. We refer to the above algorithm as DEND since it uses the dendrogram to choose the representative templates. The algorithm selects prototypes that represent the *variability* observed in a user's data. Therefore, this algorithm is prone to selecting outliers.

Method 2 (MDIST): The second method sorts the fingerprint impressions based on their *average* distance score with other impressions, and selects those impressions that correspond to the K smallest average distance scores. Here, the rationale is to select templates that exhibit maximum similarity with the other impressions and, hence, represent typical data measurements. We refer to this method as MDIST since templates are chosen using a minimum distance criteria. The prototype set selected by this technique represents data that is likely to occur frequently. Thus, for every user:

Step 1: Find the pair-wise distance score between the N impressions.

Step 2: For the jth impression, compute its average distance score, d_j, with respect to the other $(N-1)$ impressions.

Step 3: Choose K impressions that have the smallest average distance scores. These constitute the template set T.

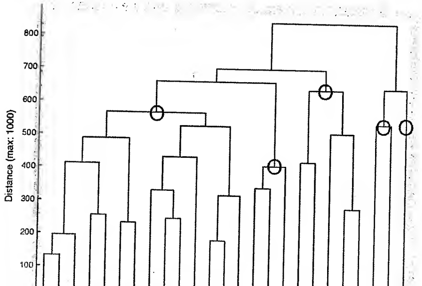


Fig. 4. Dendrogram generated using the 25 × 25 dissimilarity matrix of a finger. The circles on the subtrees indicate cluster formations for $K = 5$.

BNSDOCID: <XP_________4508402A__I_>

The choice for the value of K is application dependent. Larger K values would mean storing more templates per user, and this may not be feasible in systems with limited storage capacities. Moreover, in an identification system, matching a query (input) data with a large number of templates per user would be computationally demanding. Smaller K values, on the other hand, may not sufficiently capture the intra-class variability nor the typicality of the data, leading to inferior matching performance. Therefore, a reasonable value of K, that takes into account the aforementioned factors, has to be specified.

3. Template update

The methods presented in the previous section can be applied periodically in order to *update* the currently selected template set. In a biometric system, a user provides biometric data every time the authentication stage is invoked. Thus, newer samples of a user's biometric are made available over a period of time. This newly acquired data can be used to refresh the current template set in order to account for temporal changes that may occur in a person's biometric trait. We suggest two simple methods to perform template update using the newly acquired data.

In the first method, all current templates are replaced with templates selected from the newly acquired data set, thereby capturing temporal changes in the fingerprints (e.g., changes due to environmental conditions, sensor characteristics or subject's occupational characteristics). We call this method BATCH-UPDATE since the previously selected batch is discarded and only the newly acquired data is considered.

In the second method, both the current template set and the newly obtained data set are considered when performing template update. The template selection procedure is applied *after* augmenting the new data set with the current template set. This method is called AUGMENT-UPDATE. We provide experimental results pertaining to both these update methodologies in the next section.

The template selection and update procedure can be invoked on each user independently and is not affected by the variable number of biometric samples available for a user. Moreover, it is an off-line process that does not interfere with the real-time performance of a system. As a result, the procedure can be easily scaled to accommodate a large number of users.

4. Experimental results

In order to study the effect of automatic template selection and update on fingerprint matching, it is necessary to acquire several impressions per finger over a period of time. Standard fingerprint databases (e.g., FVC 2002 [9]) do not contain a large number of impressions per finger. Therefore, we collected 200 impressions each of 50 different fingers in our laboratory using the Identix BioTouch USB 200 optical sensor (255×256 images, 380 dpi). The data was acquired over a period of approximately four months with no more than 5 impressions of a finger per day. The 200 impressions of each finger were partitioned into two sets: the first 100 impressions (DATA1) were used to conduct the template selection experiments while the remaining 100 impressions (DATA2) were used in the template update experiments. Each of these two sets were further divided into training (first 25 impressions) and test (remaining 75 impressions) sets. These individual partitions were labelled as TRAIN1, TEST1, TRAIN2 and TEST2.

4.1. Template selection

The template selection experiments were conducted using DATA1. The selection procedure was applied to images in TRAIN1, while the matching performance was evaluated using images from TEST1.

Fig. 4 shows the dendrogram obtained using the 25 fingerprint impressions of one finger. On setting $K = 5$, the resulting clusters and their prototypes as computed using the DEND algorithm are shown in Fig. 5; some clusters are seen to have only one member, suggesting the existence of outliers. The various prototypes are observed to have different regions of overlap with respect to the extracted minutiae points. The prototypes, for the same finger, computed using the MDIST algorithm are shown in Fig. 6.

In order to assess the matching performance of the proposed techniques (for $K = 5$), we match every image in TEST1 (50 fingers, 75 impressions per finger) against the selected templates (5 per finger). When a test image is matched with the selected template set of a finger, 5 different distance scores are obtained. The mean of these scores is reported as the final matching score. [2] Thus, we obtain 187,500 matching scores ($75 \times 50 \times 50$) using the selected template sets. Fig. 7(a) shows the receiver operating characteristic (ROC) curves representing the matching performance of the template sets selected using both the algorithms. The equal error rates (EER) of DEND and MDIST are observed to be 7.42% and 6.62%, respectively. Now, for the 50 fingers, there are a total of $(\binom{25}{5})^{50} - 1$ *non-selected* template sets. It is computationally prohibitive to generate the matching scores and the ROC curves corresponding to all these permutations. Therefore, we chose 53,130 permutations (assuming that the impression indices in the template set of all the 50 fingers is the same) and computed their EER. The histogram of EER values is shown in Fig. 7(b), where the minimum, mean and maximum EER values are 6.12%, 7.89% and 10.31%, respectively. In this histogram, the vertical lines indicate the EER values corresponding to the DEND and MDIST

[2] Other techniques to combine matching scores can be found in Ref. [10].

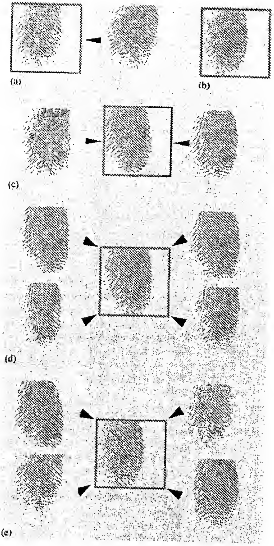


Fig. 5. The cluster membership ($K = 5$) for the dendrogram shown in Fig. 4. At most 5 members are indicated for each cluster. The prototype template in each cluster is marked with a thick border. Note that the cluster in (b) has only one member.

algorithms. The percentage of non-selected template sets that have a lower EER than the template sets selected with the proposed methods is 30.3% and 1.6%, for DEND and MDIST, respectively, thereby suggesting that systematic template selection is better than random selection.

We see that the MDIST method of template selection leads to a better matching performance compared to the DEND method of selection. This may be attributed to the fact that MDIST selects a template set consisting of images that exhibit maximum similarity with other impressions; therefore, the probability of them being correctly matched with impressions of the same finger is fairly high. On the other hand, the DEND method is prone to selecting impressions that are outliers thereby increasing the probability of false rejects. However, both methods are essential due to the complementary nature of the template set that they select.

To further understand the differences and similarities between the template sets selected by the DEND and MDIST methods, we calculated the number of common impressions selected using these two methods (for $K = 5$). The minimum, average (over all 50 users) and the maximum values for this number is found to be 0, 1.54 and 4, respectively. Since, on the average, 1.54 out of a possible 5 impressions are common in the selected sets, the difference in performance between the two techniques stems from the remaining members of the respective sets.

Table 1 lists the impressions of a finger that were selected as templates using the DEND and MDIST algorithms at different K values. The impression index indicates the acquisition time of the impressions—a lower index referring to an earlier time instance. We see that there is no direct relationship between an impression index and its choice as a template. Fig. 8 shows the EERs of the two methods at different values of K. A good choice for K (that can balance system performance with the computational/storage overheads) could be established by observing the knee point in the respective error curves (e.g., $K = 5$ for DEND). However, a more formal method needs to be developed for determining the value of K for a specific application.

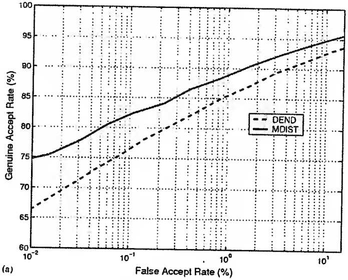
4.2. Template update

In this subsection, we report the system performance of the BATCH-UPDATE and AUGMENT-UPDATE methods. Observe that both these update techniques implicitly rely on the template *selection* procedures described earlier. They differ only in the choice of the data set on which template selection is performed. To demonstrate the importance of template update, we report the matching performance before and after the update process. We assume that the current template set of a finger consists of images from TRAIN1.

In the BATCH-UPDATE method, template selection was performed on TRAIN2 after completely discarding the template set extracted from TRAIN1. In the AUGMENT-UPDATE method, on the other hand, images from TRAIN2 were first augmented with the 5 current templates, and template selection was performed on the augmented set. Both the update techniques selected 5 templates from their respective candidate sets. The matching performance of the two update techniques was evaluated using data set TEST2. Fig. 9 shows the ROC curves indicating the performance before template update (i.e., templates selected from TRAIN1

BNSDOCID: <XP_______4508402A_I_>

Fig. 6. The prototype templates of a finger selected using the MDIST algorithm. The average distance measures for these are (a) 425, (b) 429, (c) 431, (d) 441, and (e) 452.



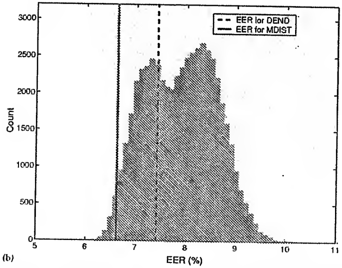


Fig. 7. (a) The ROC curves for the DEND and MDIST algorithms on a database of 50 fingers with 75 impressions per finger used in the test phase. (b) The EER histogram for the non-selected sets. The EER of the DEND and MDIST algorithms are also indicated.

Table 1
The impression indices of selected templates for a finger at different values of K

K	DEND	MDIST
1	2	2
3	4, 8, 25	2, 4, 13
5	4, 5, 9, 22, 25	2, 3, 4, 12, 13
7	2, 5, 7, 9, 19, 22, 23	2, 3, 4, 12, 13, 21, 23
9	2, 5, 7, 8, 9, 19, 20, 22, 23	2, 3, 4, 12, 13, 20, 21, 23, 25

were tested on TEST2) and after incorporating the template update procedures. Table 2 lists the EERs of both the update methods based on the selection technique (DEND and MDIST) that was employed. It is seen that both the update methods result in substantial improvement in matching performance.

We observe from the ROC curves and EER values that AUGMENT-UPDATE results in better performance than BATCH-UPDATE. This is because in AUGMENT-UPDATE we give the previously selected templates a chance to compete for reselection. This can help in retaining long-term trends in the characteristics of the fingerprint impressions leading to the observed improvement in performance. In AUGMENT-UPDATE, the average number of reselected templates (the average is taken over 50 users) was found to be 1.5 and 0.62 using the DEND and MDIST methods, respectively.

5. Discussion and future work

A systematic procedure for template selection and update is critical to the performance of a biometric system. In this paper we have proposed two techniques to perform template selection in the context of a fingerprint matching system. Both techniques are based on the distance score between pairs of fingerprint impressions originating from the same finger. The first method called DEND utilizes a clustering scheme to detect prototype impressions. The template set selected by this technique captures the variability observed in a user's fingerprint image. The second method called MDIST ranks the fingerprint impressions based on their average distance from the other impressions, and then selects impressions whose average distance is the least. Thus, it aids in selecting a template set that exhibit maximum similarity with the other impressions. Our experiments demonstrate that a systematic template selection procedure results in better performance than random template selection; it was also observed that the MDIST technique results in better performance than DEND. Currently, we are studying ways to effectively combine the two techniques in order to further improve system performance. We are also considering methods to determine the value of K automatically.

We have also proposed two template update methods that refresh the current template set based on newly acquired biometric data. The update methods implicitly rely on the template selection process. The AUGMENT-UPDATE technique performs template selection by considering both the current

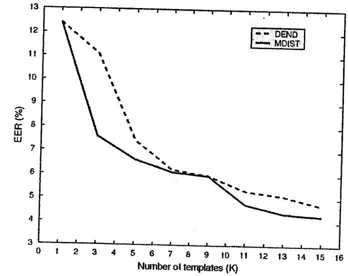
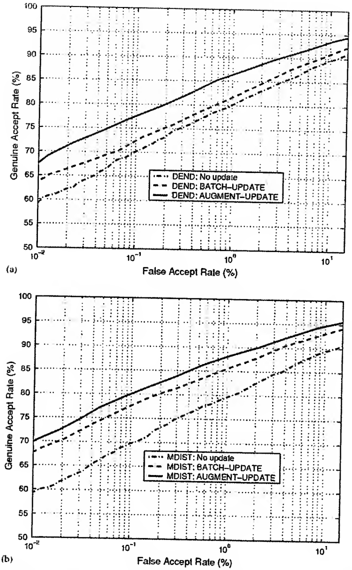


Fig. 8. The EER of the fingerprint matcher plotted as a function of K.



(a)

(b)

Fig. 9. ROC curves showing improvement in performance when the BATCH-UPDATE and AUGMENT-UPDATE procedures are incorporated. In (a) the DEND method of template selection was used, while in (b) the MDIST method was employed.

template set as well as the newly acquired data set. The BATCH-UPDATE technique, on the other hand, considers only the newly acquired data set and completely discards the current template set. Both the update methods were shown to improve the matching performance of the system. Our experiments indicate that AUGMENT-UPDATE yields better matching performance than BATCH-UPDATE.

It must be mentioned that the template selection and update techniques described here are based on the distance measure (scores) between pairs of fingerprint impressions. We have, therefore, adopted a featureless approach to clustering [11]. Hence, if a different fingerprint matching algorithm is used, a different set of prototype impressions is likely to be obtained. We are developing alternate techniques that would operate on the raw images in order to detect prototype impressions. We are also in the process of testing our techniques on other biometric modalities as well (viz., face and hand geometry).

BNSDOCID: <XP________4508402A__I_>

Table 2
EER values before and after incorporating the template update procedure

Update technique	EER (%)
DEND method	
No update	10.61
BATCH-UPDATE	9.55
AUGMENT-UPDATE	7.37
MDIST method	
No update	10.32
BATCH-UPDATE	7.69
AUGMENT-UPDATE	6.31

Acknowledgements

This project was sponsored in part by the Center for Identification Technology Research (CITeR, West Virginia)—a Industry-University Cooperative Research Center (IUCRC) funded by the National Science Foundation (NSF).

References

[1] A.K. Jain, R. Bolle, S. Pankanti (Eds.), Biometrics: Personal Identification in Networked Society, Kluwer Academic Publishers, Dordrecht, 1999.

[2] S.D. Connell, A.K. Jain, Template-based online character recognition, Pattern Recognition 34 (1) (2001) 1–14.

[3] X. Jiang, W. Ser, Online fingerprint template improvement, IEEE Trans. PAMI 24 (8) (2002) 1121–1126.

[4] K.A. Toh, W.Y. Yau, X.D. Jiang, T.P. Chen, J. Lu, E. Lim, Minutiae data synthesis for fingerprint identification applications, in: Proceedings of the International Conference on Image Processing (ICIP), Thessaloniki, Greece, Vol. 3, 2001, pp. 262–265.

[5] A.K. Jain, A. Ross, Fingerprint mosaicking, in: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Orlando, Florida, 2002.

[6] X. Liu, T. Chen, S.M. Thornton, Eigenspace updating for non-stationary process and its application to face recognition, Pattern Recognition, Special issue on Kernel and Subspace Methods for Computer Vision (2003) 1945–1959.

[7] A.K. Jain, L. Hong, R. Bolle, On-line fingerprint verification, IEEE Trans. PAMI 19 (4) (1997) 302–314.

[8] A.K. Jain, R.C. Dubes, Algorithms for Clustering Data, Prentice-Hall, Englewood Cliffs, NJ, 1988.

[9] D. Maio, D. Maltoni, R. Cappelli, J.L. Wayman, A.K. Jain, FVC2002: Fingerprint verification competition, in: Proceedings of the International Conference on Pattern Recognition (ICPR), Quebec City, Canada, 2002, pp. 744–747.

[10] J. Kittler, M. Hatef, R.P. Duin, J.G. Matas, On combining classifiers, IEEE Trans. PAMI 20 (3) (1998) 226–239.

[11] R.P. Duin, D. de Ridder, D.M.J. Tax, Experiments with a featureless approach to pattern recognition, Pattern Recognition Lett. 18 (1997) 1159–1166.

[12] R.-L. Hsu, Face detection and modeling for recognition, Ph.D. Thesis, Michigan State University, 2002.

[13] A.K. Jain, A. Ross, S. Pankanti, A prototype hand geometry-based verification system, in: Second International Conference on Audio and Video-based Biometric Person Authentication (AVBPA), Washington, DC, USA, 1999, pp. 166–171.

About the Author—UMUT ULUDAG received B.S. and M.S. degrees in Electrical and Electronics Engineering from Bogazici University, Istanbul, Turkey in 1999 and 2001, respectively. He was a researcher in Marmara Research Center, Turkey, from 1999 to 2001. He is now a Ph.D. student in Department of Computer Science and Engineering, Michigan State University, East Lansing, MI. His research interests include biometrics, pattern recognition, watermarking, multimedia, image processing and computer vision. He is a member of the IEEE.

About the Author—ARUN ROSS is an Assistant Professor in the Lane Department of Computer Science and Electrical Engineering at West Virginia University. Ross received his B.E. (Hons.) degree in Computer Science from the Birla Institute of Technology and Science, Pilani (India), in 1996. He obtained his M.S. and Ph.D. degrees in Computer Science and Engineering from Michigan State University in 1999 and 2003, respectively. Between July 1996 and December 1997, he worked with the Design and Development group of Tata Elxsi (India) Ltd., in Bangalore. He also spent three summers (2000–2002) with the Imaging and Visualization group at Siemens Corporate Research, Inc., Princeton, working on fingerprint recognition algorithms. His research interests include statistical pattern recognition, image processing, computer vision and biometric authentication.

About the Author—ANIL JAIN is a University Distinguished Professor in the Departments of Computer Science & Engineering and Electrical and Computer Engineering at Michigan State University. His research interests include statistical pattern recognition, exploratory pattern analysis, Markov random fields, texture analysis, 3D object recognition, medical image analysis, document image analysis and biometric authentication. Several of his papers have been reprinted in edited volumes on image processing and pattern recognition. He received the best paper awards in 1987 and 1991, and received certificates for outstanding contributions in 1976, 1979, 1992, 1997 and 1998 from the Pattern Recognition Society. He also received the 1996 IEEE Transactions on Neural Networks Outstanding Paper award. He is a fellow of the IEEE, ACM, and International Association of Pattern Recognition (IAPR). He has received a Fulbright Research Award, a Guggenheim fellowship and the Alexander von Humboldt Research Award. He delivered the 2002 Pierre Devijver lecture sponsored by the International Association of Pattern Recognition (IAPR). He holds six patents in the area of fingerprint matching.

BNSDOCID: <XP_______4508402A__I_>